# Improving multi-level interactions modelling in a multi-agent generalisation model: first experiments

A Maudet, Guillaume Touya, C Duchêne, S. Picault

# Improving multi-level interactions modelling in a multi-agent generalisation model: first experiments

A. Maudet[1], G. Touya[1], C. Duchêne[1], S. Picault[2]

[1]Université Paris Est, IGN, COGIT, Saint-Mandé, France
Email: {adrien.maudet, guillaume.touya, cecile.duchene}@ign.fr

[2]Laboratoire d'Informatique Fondamentale de Lille (LIFL), Université Lille 1, France
Email: sebastien.picault@lifl.fr

## 1. Introduction

This paper introduces a beginning PhD study seeking to improve the modelling of multi-level interactions between cartographic objects within a generalisation process. What we call "interaction" between two objects is the fact that the transformation of one object is computed while considering the other one (*e.g.* a building is displaced away from another one, the new position of a point is computed to solve equations in which other points are involved). Interactions translate the contextual nature of generalisation. Our aim is to overcome remaining issues linked to multi-level interaction not addressed by current approaches in automated generalisation.

The remainder of the paper is structured as follows. Section 2 describes the motivations for this work. Section 3 introduces a recent interactions-oriented multi-level model from the multi-agent literature and its adaptation to generalisation. Section 4 describes first experimentations on a case study. Finally, section 5 concludes and draws some perspectives of further work.

## 2. Motivation – identifying unsolved problems

Before exposing the unsolved problems the PhD study focuses on, we analyse some existing approaches using a typology based on the levels of interactions between objects. By level, we mean space into which it is coherent for objects to interact in order to contribute to the generalisation process (*e.g.* the building of a same block, the roads of a same network).

Among existing approaches of automated generalisation, some are based on *transversal interactions*, i.e. interactions between objects of the same level. For instance, Aslan et al. (2012), or Duchêne et al. (2012) in the CartACom model, solve conflicts between single objects by moving them away from each others. Continuous optimisation approaches express intra- or inter-objects constraints as equations on their points and then base their resolution on interactions between points (Højholt 1998, Harrie and Sarjakoski 2002, Sester 2005). Other approaches explicitly model several levels of objects and enable *hierarchical interactions*. The AGENT model (Ruas 1999) represents composition relationships between entities called components (*e.g.* buildings), and groups called meso objects (*e.g.* urban blocks). A component may be a single entity called micro, or another meso. Such relations are called composition relationships since the meso object is composed of its components. The micro and meso objects are modelled as autonomous agents that apply generalisation algorithms to themselves. Meso agents trigger their components and can impose some transformations to them to solve shared conflicts. The GAEL model (Gaffuri et al. 2008) adds the notion of field agents, representing single objects or background objects like the relief, which can dynamically decompose themselves into a set of

points (their vertices) and then orchestrate their iterative displacement under constraints. The interactions are *hierarchical* (field/points) and *transversal* (e.g. the points trigger each other).
With these existing approaches, a generalisation process can use:

- *Transversal interactions* between points (GAEL, continuous optimisation) or between single objects (CartACom).
- *Hierarchical interactions* between objects linked by a composition relationship (AGENT, GAEL where the composition is from an object to its vertices).

A few attempts have been made to sequentially use both kinds of interactions. In GAEL, the fields activate their points (hierarchical interactions) in response to a transversal interaction, but the issue of "when to switch from the transversal to the field-points interactions?" is not solved. The CollaGen model (Touya et al. 2011) enables to sequentially use several models (and thus several kinds of interactions), but while considering each process as a black box, so that one object is only involved in one interaction scheme at a time.

We consider it is not enough. First, because on top of "composition", which is well represented in the AGENT and GAEL model, "inclusion" is a second kind of hierarchical relationships that drives contextual generalisation, as identified by Mustière and Moulin (2002). Inclusion relationships notably occur when one object acts as "spatial support" for objects of equal or lower dimension that have to stay within it, as defined by Jaara et al. (2012), e.g. bus stops or accidents on a road. The issues implied by this kind of relationship cannot be solved in an easy way, like adding the bus-stop, or the accident, to a vertex of the road. Indeed, the position of objects is more dependant on the "semantic" description of the road (*e.g.* "in a bend", "near a junction") than the geometric one. Second, because in either cases (composition or inclusion), it can be necessary to use more than only hierarchical or only transversal interactions (Figure 1):

- "Diagonal" interactions (between objects at different levels) can be needed, e.g. when a group of two adjacent buildings needs to interact as a whole with a neighbouring road or building (Figure 1a). The word "diagonal" is used, since we consider links between the two buildings and the whole as "vertical" (between objects of different levels with a hierarchical relation), and links between buildings (both aggregated buildings and the others) as "horizontal" (within one level).
- An object involved in a hierarchy may need to hierarchically interact with its "parent" but also transversally interact with its pairs, *e.g.* a bus station should stay on the road when the road's shape is modified, but also remain consistent with other bus stations (Figure 1b).
- An object can be involved in more than one hierarchical relationship, *e.g.* a building can belong to two rows of buildings (Figure 1c) or a bridge is included in both a road and a river (Figure 1d).

Currently, these issues are not solved in a satisfying way. The existing models are used on the portions where they are effective and adjustments are manually done for the resulting conflicts. A satisfying result would be to have a completely automated process that orchestrates the interactions between agents from all levels.

Our hypothesis is that a generic modelling of multi-level interactions should help in addressing these issues in a more comprehensive way. Multi-level modelling has recently been intensively studied in the multi-agents domain. It resulted in several models, among which one, called PADAWAN, seems to have very interesting properties in our context. Therefore, we will try to improve the current AGENT, GAEL and CartACom agent approaches of generalisation with this model.

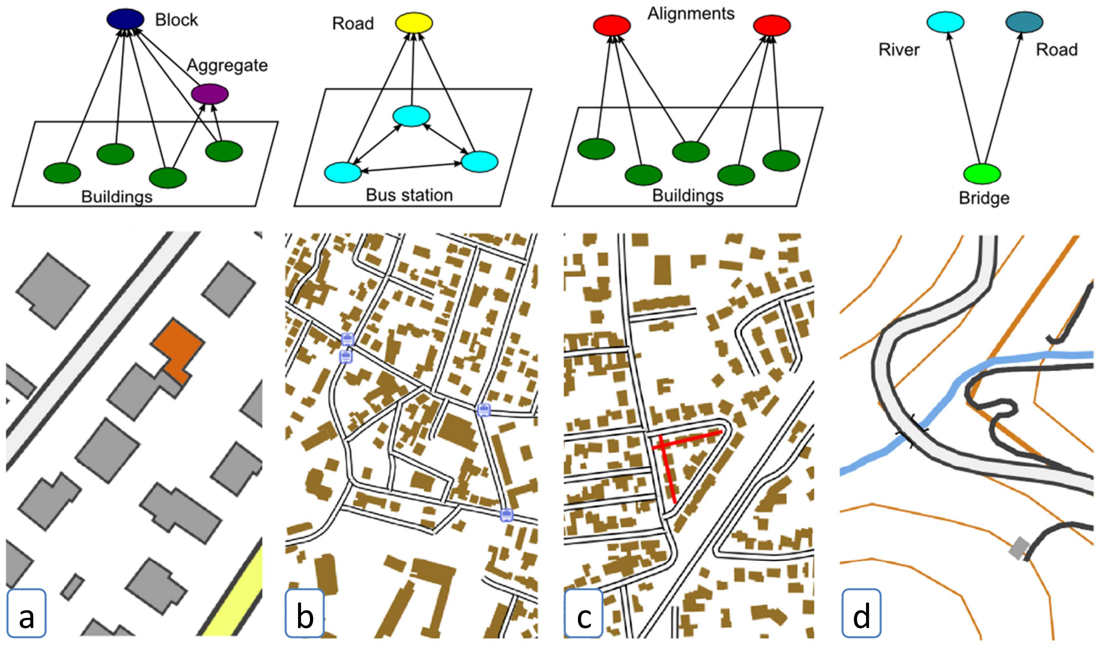



Figure 1. Some cases where the alternative hierarchical OR transversal interactions is not sufficient.

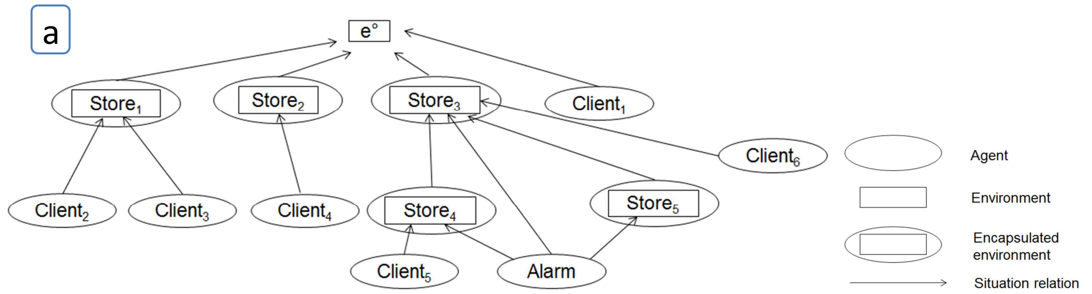
# 3. Reusing a multi-agent framework designed for multi-level simulation

## 3.1 PADAWAN: general presentation

In this section, we present the PADAWAN model (Picault and Mathieu 2011). First, it allows agents to encapsulate environments, (either physical or social) which can themselves host lower-level agents (*e.g.* a lift is an agent moving between floors, and an environment hosting other agents). This can be used to represent geographical objects acting as a spatial support for others (such as a road which "includes" bus stations), or composition relations between objects (such as groups of buildings which can be reified by specific meso agents). Then, any agent can be situated in several environments at the same time. Thus the structure of the multi-level hierarchy can be more complex than a mere tree and allows for instance the modelling of a bridge belonging to both a road and a river. Figure 2a shows the organisation of a shopping arcade instance, with the PADAWAN model: stores are both environments in which clients, alarms and other stores may be situated, and agents situated themselves in environments ($e^0$ represents the top level environment in which all agents are embedded).

Finally, the behaviour representation in PADAWAN relies upon an interaction-based model called IODA (Kubera et al, 2011), where interactions are condition/action rules involving several agents. The contextual variations in agents behaviours, depending on the environment they are situated in, is expressed by an interaction matrix (what interactions can occur between pairs of agent families) attached to each environment as shown in Figures 2b and 2c. This could help represent that *e.g.* buildings would not apply the same generalisation rules in city center blocks and suburban blocks.

Yet, the PADAWAN model was designed for multi-agent simulation: it has to be adapted to the context of generalisation which is not simulation but spatial problem solving.




b

| Source/ Target | Ø | Store as Host | Client | Store | Item | Alarm |
|---|---|---|---|---|---|---|
| Store as Host | | | Inform | Open Close | SellOff Remove | SwitchOf |
| Client | Explore | Pay Exit | | Enter | Take | Activate |
| Store | | Pay% | Attract | | | |
| Alarm | Ring | Alert | | | | |

c

| Source/ Target | Ø | Store as Host | Client | Store | Item | Alarm |
|---|---|---|---|---|---|---|
| Store as Host | | | Inform | Open Close | SellOff Remove | SwitchOf |
| Client | Explore | Pay Exit | | Enter | **Order** | Activate |
| Store | | Pay% | Attract | | | |
| Alarm | Ring | Alert | | | | |

Figure 2. A hierarchical architecture in the PADAWAN model and instances of interaction matrix. (a) rectangular shapes indicate environments and oval shapes agents. An arrow indicates a location relation. For instance, *Client$_1$* is located in *Store$_1$*. (b) the interaction matrix for environment encapsulated by *Store$_1$*, *Store$_3$* and *Store$_4$*. The one indicated by (c) is for *Store$_2$* and *Store$_5$* and include a different interaction.

### 3.2 Integration for AGENT: adaptation and implementation

As a first step of our work, we focus on the adaptation of the AGENT model into the PADAWAN paradigm seeking to obtain the same results. We express the algorithms used by AGENT into PADAWAN interactions. In AGENT, constraints propose actions to be solved (e.g. a “change into smallest surrounding rectangle” to solve a “squareness” constraint). In PADAWAN, the interactions are allowed by so called preconditions (conditions necessary for an interaction to be possible) and motivated by so called triggers (conditions that motivate the interaction), *e.g.* for a client agent, the “Take” interaction with an item agent is allowed by the “have enough money” precondition and the “need this item” trigger. The influence of constraints on the triggering of possible actions has to be expressed in PADAWAN. In this purpose, we introduce the notion of constraint advice. A constraint may express a *favourable*, *unfavourable*, *indifferent* or *opposite* advice on the application of an action on the object. According to the advices of all constraints and the unsatisfaction of some of these constraints, the model decides if an action will be triggered, launched with reluctance, or not launched at all (because it is likely that the execution of the action will have negative and hardly reversible effects on the result).
Once the actions to execute are chosen, they are ordered according to several factors: the importance for the constraint to be solved and the degree of unsatisfaction of the constraint. Each action is tested on the object, and if it effectively enhances the satisfaction degree, is validated.

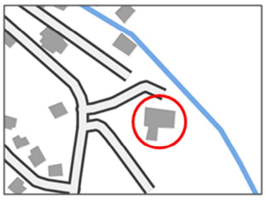
Using this model, we express the interactions of encapsulated by an urban block in Figure 3).

| Source/ Target | Ø | Block as a Host | Building |
|---|---|---|---|
| Block as a Host | Transform in town center | | Suppress<br>Displace<br>Activate |
| Building | Suppress<br>Enlarge<br>Rotate<br>Simplify<br>Replace by rectangle | | |

Figure 3. An interaction matrix for a block environment. The block as a host may interact on himself in one way (when transforming into town center, *i.e.* being filled in a single colour), or on its buildings (when suppressing or displacing some of them, or when activating them). When activated, a building may do some operation on itself.

As it appears in the matrix, there are only hierarchical (host/building cell) and reflexive (Ø column) interactions. The purpose of PADAWAN is to take into account relations including all kinds of agents, for instance building/ building interactions. This other kinds of relation will be used when adapting CartACom and GAEL models to PADAWAN.

### 3.3 Implementation and results



| | |
|---|---|
| Initial State S=94.8<br>→ **Simplification S=98.8**<br>→ Affinity S=88.3<br>→ Squaring S=94.8<br>→ SSR S=93.3 | Initial State S=94.8<br>→ **Simplification S=98.8**<br>→ Affinity S=88.3<br>→ Squaring S=94.8<br>→ SSR S=93.3 |

Without PADAWAN With PADAWAN

Figure 4. Comparisons of the AGENT model, without and with PADAWAN. The same results are obtained.

CartAGen is a platform implementing some generalisation algorithms, including the AGENT model (Renard et al, 2010). Our proposition is implemented in this platform and the results are compared with those of the original AGENT model (Figure 4). The global satisfaction value (S) is computed from the satisfaction of each constraint. Each action are proposed and tested in a given order. S is used to validate the effectiveness of an action after trying it. If S decreases, the action is cancelled. The map at left shows the building before generalisation. If S increases, new possible actions are computed and tested. Test carried out show that: with the PADAWAN translation of the AGENT model, we obtain the same result as with the native version of the AGENT model.

We remind that the purpose of this first step was to adapt an existing model to another one. The expected improvements were not on the efficiency, nor the effectiveness, but on the flexibility to manage multi-level interactions. The results prove that the modified PADAWAN model is as good as the former AGENT model. The next section shows with a case study that it is additionally a good way to model diagonal interactions.

# 4. Case study: the dead-end streets and their neighbourhood

The section focuses on the generalisation of dead-end streets and their neighbourhood inside an urban generalisation by AGENT. After the problem being presented, new generalisation operations are defined, and then integrated into the adapted PADAWAN model.

## 4.1 Presentation of the case study

During the generalisation process, road symbols are enlarged to fit the perception threshold. This enlargement may reduce the space between two road sections, and prevent other objects like buildings to find room to display. This problem especially occurs with dead-end streets (Figure 5).

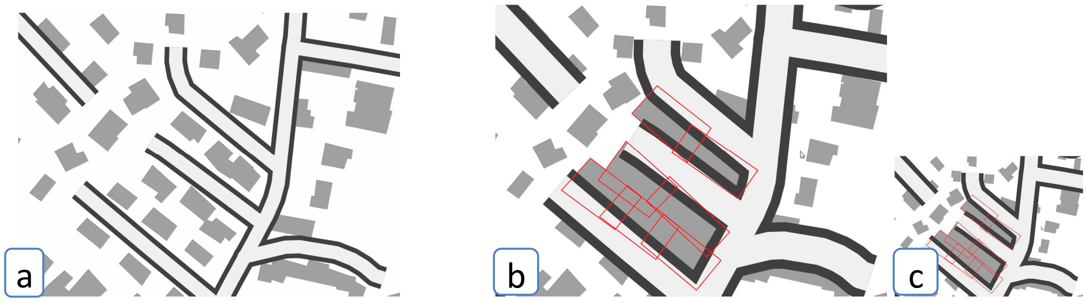



Figure 5. (a) a map portion including dead-ends street at 1:25k symbolisation scale. (b) the same portion at 1:50k symbolisation scale with road symbols enlarged and related buildings (surrounded in red) magnified to reach minimum size at 1:50k. (c) same as (b) displayed smaller.

Whit the current AGENT model, to solve this sort of problem, the more common solution when there is not even enough space for one row of building is to suppress the dead-ends first and, if necessary, the buildings. But a better way may be to displace dead-ends to give enough space to the buildings. This displacement will impact the structure of the dead-ends' neighbourhood. To maintain consistency with reality, we need to move objects in the neighbourhood of the dead ends. Also, when a dead-end is moved, its junction node with the network is moved too. This displacement may cause inconsistencies with other elements located on the road, like bus stations. Of course, these objects continue to be hosted in other environments, and they have to manage the previously existing constraints besides the new ones (*e.g.* the buildings are always in a block, so they need to respect a constraint of density carried by the block; the bus station needs to preserve its relative position to other stations within the road).

The next section presents how to model this constraint based dead-end displacement problem.

### 4.2 Modelling of the problem

The problem may be modelled in different ways. First, we describe a modelling with only vertical and horizontal relations based on the principle of AGENT and CartACom, and second we propose a modelling with a more complex multi-level architecture using our modified PADAWAN model.

A simple way is to model the dead-end road as an agent in a block, as shown in Figure 6a. This agent interacts with neighbouring buildings. This may be either horizontally and directly in a CartACom way, or vertically and through a meso agent in an AGENT way. However, we add complexity to existing models with the addition of a new type of objects in the urban block. This raises a problem to calibrate the constraints and interactions, which is already difficult in the original approaches.

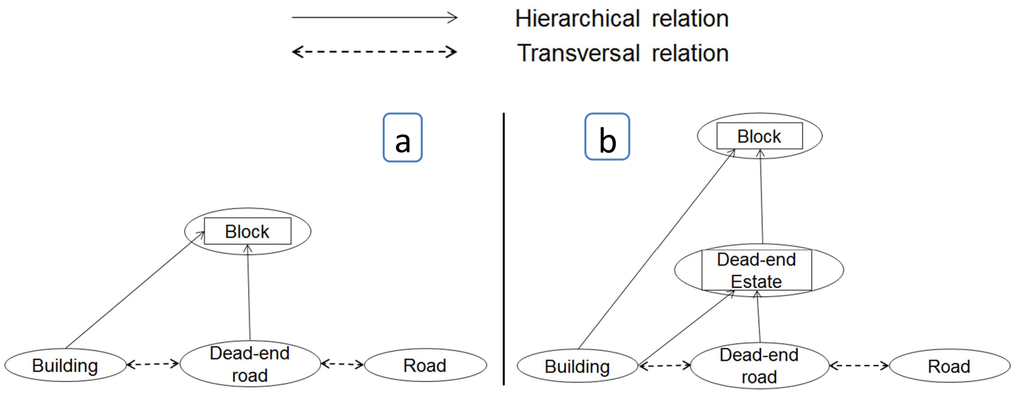



Figure 6. Two way of modelling the problem.

Instead, a model for the organisation of the objects involved in the problem, as PADAWAN environments and agents is proposed (Figure 6b). A first step is to define the dead-end as an object type in our model. We do not only consider the dead-end as a road, but we define a “dead-end estate” agent that encapsulates a PADAWAN environment including all the building reachable by the dead-end road, and the dead-end road itself. This environment is created using a buffer delimiting a surface around the dead-end road. All buildings overlapping the buffer are considered as included in the dead-end estate. The purpose of this dead-end estate agent is to consider it as a whole for some operations, but the objects situated inside need to be considered as individuals for some other operations. It is an instance of multi-level situation with “diagonal” interactions, since the dead-end estate is considered as an element of the block level, as the buildings inside it. Figure 7 shows an instance of encapsulation relations in a block with dead-ends.

Then, two constraints are defined (one for each side of the dead-end) to detect if there is enough room on the left resp. on the right for at least one row of buildings. If the value is too small (compared to a minimal value calculated on the mean size required by a typical building and on the width of the road), the constraint is unsatisfied. A ray-tracing strategy is used to detect objects on the left resp. right of a section: for some points of the section, a ray orthogonal to the section is traced. This indication gives us a way to evaluate the satisfaction of each of the two “is there enough room on this side of the dead-end” constraint. Those constraints are carried by the previously define dead-end estate agent.

Having the left (resp. right) constraint unsatisfied is a trigger (in the PADAWAN sense) for a “slide to right” (resp. “slide to left”) action, which

- Slides the dead-end road toward the direction opposite here there is not enough room
- Moves the buildings hosted by the dead-end estate by translating them with the same displacement vector.

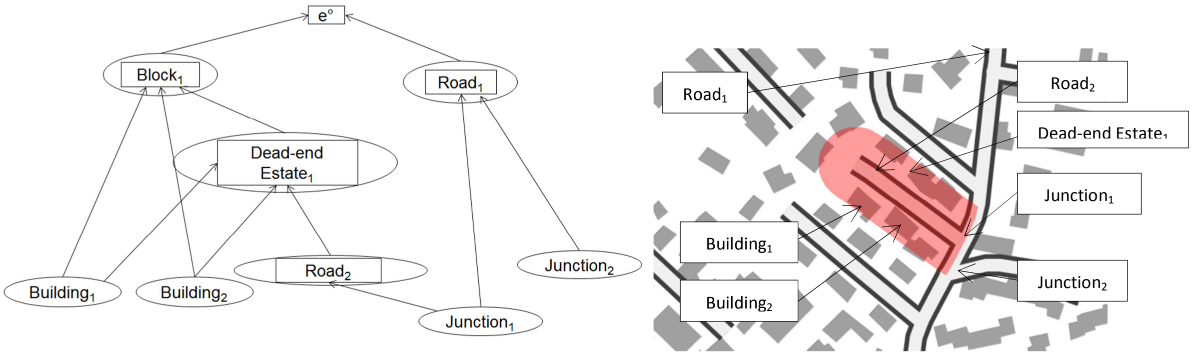



Figure 7. Definition of the organisation of PADAWAN environments and agents.

The priority for this constraint to be solved is high because room needs to be freed to allow buildings to be generalised correctly. But this approach does not model the need to respect information consistency on the roads. Therefore, to avoid inconsistencies, for now, a displacement is not allowed when a junction with another road is detected close to the dead-end junction, so that the order of junctions with dead-ends along a road cannot be accidentally modified.  All that information is expressed as interactions and integrated in the matrix associated to the environment encapsulated by the block (Figure 8).
To complete the process, an interaction to order and activate the dead-end estate agent in a block environment, is needed. This interaction is carried by the block agent. The dead-end estate agents to activate are sorted depending on the distance of the dead-end to a non-dead-end road.

| Source/Target | Ø | Block as a Host | Building | Dead-end estate |
|---|---|---|---|---|
| Block as a Host | Transform in town center | | Suppress Displace Activate | Activate |
| Building | Suppress Enlarge Rotate Simplify Replace by rectangle | | | |
| Dead-End estate | Slide to left Slide to right | | | |

Figure 8. Modified interaction matrix from Figure 3 including a new agent family and associated interactions.

The complexity of this modelling is higher than the one from original AGENT solutions, but remains in the same proportions. In future works, new constraints will be implemented, including more computing and complex situations. One of the aim of the PhD study is to manage this complexity using mechanisms of the new model.

### 4.3 Results

The proposed solutions have been implemented in the CartAGen platform. The new agent type "dead-end estate" has been defined. New algorithms, new interactions and associated interactions matrix has been implemented. Figure 9 shows results obtained in an urban area, with displaced dead-end on the left of the figure. These results are more interesting than the previous way consisting in deleting all small dead-ends.


Figure 9. Instance of displacement of dead-end streets for a 1:50k map. (a) the original data (1:25k). (b) the generalized data for a 1:50k map. (c) same as (b) displayed smaller.

## 5. Conclusion and perspectives

Some unsolved generalisation problems were identified as multi-level issues that require to go further than the existing multi-agent models. To propose a solution, we assume that a more formalised way to express interactions between agents in multi-level context may be a first step to solve them. To do that, we choose the PADAWAN model from the MAS literature. As a first step of our demonstration of the usability of PADAWAN as a model for cartographic generalisation, the AGENT model has been adapted to the PADAWAN model. This adaptation needed adjustments and enhancements to PADAWAN.

The dead-end streets case study is our first concrete new problem. To solve it, a model for dead-ends street agents and algorithms was proposed and integrated in PADAWAN model. The first results show better generalisation of blocks with dead-end streets.

A next step is to consider the problem integrating the order of elements on roads and propose an understandable way to describe the orchestration of the agents with multi-level interactions. After that, the objectives are to solve other problems, including those detailed in part 2 and to fusion the three COGIT MAS for generalisation: AGENT, CartACom and GAEL relying on the enhanced PADAWAN paradigm.

## References

Aslan S, Bildirici I, Simav Ö and Çetinkaya B, 2012, An Incremental Displacement Approach Applied to Building Objects in Topographic Mapping. In: *Proceedings of 15th ICA Workshop on Generalisation and Multiple Representation*, Istanbul, Turkey.

Duchêne C and Gaffuri J, 2008, Combining three multi-agent based generalisation models: AGENT, CartACom and GAEL. In: *Proceedings of 13th International*

*Symposium on Spatial Data Handling (SDH'08)*, 23-25 June, Montpellier (France), 277–296.

Duchêne C, Ruas A and Cambier C, 2012, The CartACom model: transforming cartographic features into communicating agents for cartographic generalisation. *International Journal of Geographic Information Science*, 26(9):1533–1562.

Gaffuri J, Duchêne C and Ruas A, 2008, Object-field relationships modelling in an agent-based generalisation model. In: *Proceedings of 11th ICA workshop on Generalisation and Multiple Representation*, Montpellier, France.

Harrie L and Sarjakoski T, 2002, Simultaneous Graphic Generalisation of Vector Data Sets. *GeoInformatica*, 6(3):233–261.

Højholt P, 1998, Solving Local and Global Space Conflicts in Map Generalisation Using a Finite Element Method Adapted from Structural Mechanics. In: *Proceedings of Spatial Data Handling*, Vancouver, Canada, 679–689.

Jaara K, Duchêne C and Ruas A, 2012, A model for preserving the consistency between topographic and thematic layers throughout data migration. In: *15th International Symposium on Spatial Data Handling (SDH'12)*, Bonn, Germany.

Kubera Y, Mathieu P and Picault S, 2011, IODA: An Interaction-Oriented Approach for Multi-Agent Based Simulations. In: *Journal of Autonomous Agents and Multi-Agent Systems*, Springer, 23(3): 303–343.

Picault S and Mathieu P, 2011, An Interaction-Oriented Model for Multi-Scale Simulation. In: *22nd International Joint Conference on Artificial Intelligence (IJCAI/AAAI)*, Barcelona, Spain, 332–337.

Renard J, Gaffuri J and Duchêne C, 2010, Capitalisation problem in research – example of a new platform for generalisation: CartAGen. In: *12th ICA Workshop on Generalisation and Multiple Representation*, Zürich, Switzerland.

Ruas A, 1999, Modèle de généralisation de données géographiques à base de contraintes et d'autonomie. PhD Thesis, Université Marne La Vallée.

Sester M, 2005, Optimization approaches for generalization and data abstraction. *International Journal of Geographic Information Science*, 19(8-9):871–897.

Touya G and Duchêne C, 2011, CollaGen: Collaboration between automatic cartographic generalisation processes. In: Ruas A (ed), *Advances in Cartography and GIScience*, LNG&C, Springer, Berlin, Heidelberg, 541–558.